

Quick  
SearchStandard  
SearchAdvanced  
SearchExtended family  
SearchCitation  
SearchSimilarity  
Search

F

[Back](#) | [Update Info](#) | [Hit List](#) | [View Records](#) | [Refine search](#) | [Shopping basket](#) | [Scroll session](#) | [SubAccount](#) | [Logout](#)

Database: PLUSPAT - Worldwide patents Results : 1 Quick Search : (DE19859389/PN)

Analyze Top: 


F

[◀◀ First](#) [◀ Previous](#) 1 / 1 [Next ▶](#) [Last ▶▶](#)

Select A



DE19859389 - (C1) Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor...

[View Extended Family](#) | [Visualize Citations](#)Patent Number:  DE19859389 C1 20000706 [DE19859389]Other Title: (C1) Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor und Prozes-  
zur Durchführung des Verfahrens

Patent Assignee: (C1) SYSTEMONIC AG (DE)

Inventor(s): (C1) WEIS MATTHIAS (DE)

Application Nbr: DE19859389 19981222 [1998DE-1059389]

Priority Details: DE19859389 19981222 [1998DE-1059389]

Intl Patent Class: (C1) G06F-009/28

EPO ECLA Class: **G06F-009/318T**

Document Type: Basic

Publication Stage: (C1) Patent Specification (First publication)

**Abstract:** The invention relates to a method for controlling functional units in a processor, according to a configuration a sequence of primary instruction words which consists of several instruction word parts and originates from a translation of a program code is compressed and stored as associated program words. The invention also relates to a processor system for carrying out the method. The aim of the invention is to increase operating speed in an application-specific processor while retaining a low program word width. To this end, as regards the method, a program word contains a first characteristic of a primary instruction word and instruction word parts which differentiate the primary instruction word belonging to the program word from the primary instruction word belonging to the characteristic. By means of the first characteristic contained in the primary instruction word a secondary instruction word is generated by exchanging the instruction word parts of the primary instruction word with those in a preceding secondary instruction word. On the system of the invention is solved by providing for the instruction word buffer to consist of a memory with optional line-by-line access.

Update Code: 2000-26

1 / 1 PLUSPAT - ©

[◀◀ First](#) [◀ Previous](#) 1 / 1 [Next ▶](#) [Last ▶▶](#)



21 Aktenzeichen: 198 59 389.9-53  
22 Anmeldetag: 22. 12. 1998  
43 Offenlegungstag: -  
45 Veröffentlichungstag  
der Patenterteilung: 6. 7. 2000

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

73 Patentinhaber:  
Systemonic AG, 01099 Dresden, DE

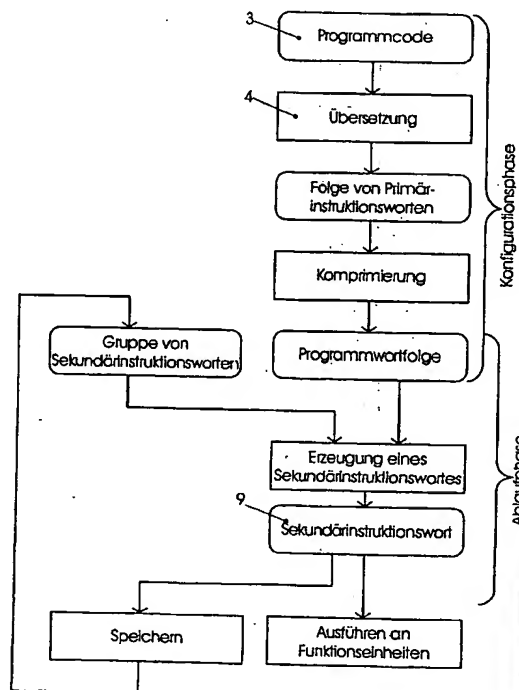
74 Vertreter:  
Patentanwälte Lippert, Stachow, Schmidt &  
Partner, 01309 Dresden

72 Erfinder:  
Weiß, Matthias, Dipl.-Ing., 01187 Dresden, DE

56 Für die Beurteilung der Patentfähigkeit in Betracht  
gezogene Druckschriften:  
Weiss, H., Fettweis, G.: "Dynamic Codewidth Reduc-  
tion for VLIW Instruction Set Architectures in  
Digital Signal Processors", Tagungsbericht des 3.  
Int. Workshop on Signal and Image Processing  
IWSIP  
1996, S. 517-520;

54 Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor und Prozessoranordnung zur  
Durchführung des Verfahrens

57 Der Erfindung, die ein Verfahren zur Ansteuerung von  
Funktionseinheiten in einem Prozessor, bei dem in einer  
Konfiguration eine aus einer Übersetzung eines Pro-  
grammcodes stammende Folge von Primärinstrukti-  
onsworten aus mehreren Instruktionswortteilen komprimiert  
und als Folge von zugehörigen Programmworten gespei-  
chert wird, und eine Prozessoranordnung zur Durchfüh-  
rung des Verfahrens betrifft, liegt die Aufgabe zugrunde,  
unter Beibehaltung einer geringen Programmwortbreite  
die Arbeitsgeschwindigkeit anwendungsspezifisch zu er-  
höhen. Verfahrensseitig wird dies dadurch gelöst, daß ein  
Programmwort ein erstes Kennzeichen eines Primärin-  
struktionswortes und Instruktionswortteile, die das zu  
dem Programmwort gehörige Primärinstrukti-  
onswort unterscheidenden, enthält. Mittels dem in dem  
Programmwort enthaltenen ersten Kennzeichen wird ein  
Sekundärinstrukti-  
onswort erzeugt, indem aus einem vor-  
hergehenden Sekundärinstrukti-  
onswort die in dem Pro-  
grammwort enthaltenen Instrukti-  
onswortteile aus-  
getauscht werden. Anordnungs-  
seitig wird die Aufgabe da-  
durch gelöst, daß der Instrukti-  
onswortpuffer aus einem  
Speicher mit wahlweisem zeilenweisen Zugriff besteht.



Die Erfindung betrifft ein Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor, bei dem in einer Konfiguration eine aus einer Übersetzung eines Programmcodes stammende Folge von Primärinstruktionsworten aus mehreren Instruktionsteilen komprimiert und als Folge von zugehörigen Programmworten gespeichert wird. Dabei werden in einer nachfolgenden Ausführungsphase aus der Folge von Programmworten aufeinanderfolgende Sekundärinstruktionsworte aus mehreren Instruktionsteilen in der zur Ansteuerung aller Funktionseinheiten benötigten vollen Instruktionsteilbreite erzeugt.

Die Erfindung betrifft auch eine Prozessoranordnung zur Durchführung des Verfahrens mit Funktionseinheiten, einem diesen Funktionseinheiten zugeordneten Instruktionsteilwortspeicher und einem Instruktionsteilpuffer zur Speicherung bereits erzeugter Instruktionsteile mit einer Breite, die mindestens gleich der Bitbreite des Sekundärinstruktionsteilwortes ist.

Prozessoranordnungen der eingangs genannten Art beinhalten Funktionseinheiten, die parallel zueinander arbeiten und die zu jedem Takt von einem Instruktionsteilwort angesteuert werden. Das jeweilige Instruktionsteilwort wird aus einem Programmwort gewonnen, das einem Programmspeicher entnommen wird.

Die Instruktionsteile ihrerseits bestehen aus mehreren Instruktionsteilworten, wobei je ein Instruktionsteilwort der Ansteuerung je einer Funktionseinheit dient.

Zur Erhöhung der Leistungsfähigkeit von Prozessoranordnungen besteht das Bestreben nach einer Vergrößerung der Verarbeitungsbreite, was eine Erhöhung der Anzahl von Funktionseinheiten erforderlich werden läßt. Damit vergrößert sich im allgemeinen die Bitbreite der Instruktionsteile und damit auch der Programmworte. Dies hat eine Bereitstellung eines entsprechenden Speicherplatzes im Programmspeicher zur Folge, der der wesentliche Flächenanteil auf dem Halbleiterchip einnimmt.

Da die Größe der Programmworte die Größe des Programmspeichers bestimmen, ist es das Bestreben, die Größe der Programmworte zu verringern, um eine Verringerung des Speicherplatzbedarfes zu erreichen. Hierfür sind mehrere Komprimierungsverfahren bekannt. Das naheliegendste Verfahren ist in dem Bericht von Weiss H., Fettweis, G. "Dynamic Codewidth Reduction for VLIW Instruction Set Architectures in Digital Signal Processors" (Tagungsbericht des 3. Int. Workshop on Signal and Image Processing IWSIP '96 S. 517 bis 520) beschrieben. Darin werden die Programmworte aus hintereinanderliegenden Primärinstruktionsteilworten so zusammengestellt, daß anschließend daraus wieder Sekundärinstruktionsteile erstellt werden können, indem ein einmal erzeugtes Sekundärinstruktionsteilwort (VLIW) in einen Instruktionsteilwortspeicher geschrieben wird und zur Erzeugung des nächsten Sekundärinstruktionsteilwortes bei dem gespeicherten Sekundärinstruktionsteilwort nur die Instruktionsteile ausgetauscht werden, die sich von dem gespeicherten Sekundärinstruktionsteilwort zu dem zu erzeugenden Sekundärinstruktionsteilwort unterscheiden. Mithin braucht in dem Programmwort nur die Information enthalten zu sein, welches Instruktionsteilwort sich unterscheidet und mit welchem Inhalt es sich unterscheidet. Damit ist es möglich, die Programmworte sehr schmal zu gestalten und damit Speicherplatz zu sparen.

Bei größeren Unterschieden zwischen dem gespeicherten und dem herzustellenden Sekundärinstruktionsteilwort ist allerdings die Breite des Programmwortes zu vergrößern, wenn diese größeren Unterschiede häufig auftreten, was den Nachteil eines größeren Speicherplatzes mit sich bringt,

oder die Unterschiede sind über mehrere Programmwörter zu verteilen. Damit ist das Sekundärinstruktionsteilwort aus mehreren Programmworten über mehrere Takte herzustellen. Daraus entsteht der Nachteil eines größeren Zeitaufwandes.

Der Erfindung liegt damit die Aufgabe zugrunde, unter Beibehaltung einer geringen Programmwortbreite die Arbeitsgeschwindigkeit anwendungsspezifisch zu erhöhen.

Verfahrensseitig wird die Aufgabe dadurch gelöst, daß im Ergebnis der Konfiguration ein Programmwort ein erstes Kennzeichen eines Primärinstruktionsteilwortes aus einer ersten Gruppe vorhergehender Primärinstruktionsteile, welches mit dem zum Programmwort zugehörigen Primärinstruktionsteilwort die größte Übereinstimmung aufweist, und Instruktionsteile, die das zu dem Programmwort gehörige Primärinstruktionsteilwort von dem zu dem Kennzeichen gehörigen Primärinstruktionsteilwort unterscheidenden, enthält. In der Ausführungsphase wird eine der ersten Gruppe in der Anzahl entsprechende zweite Gruppe von Sekundärinstruktionsteilen mit je einer zweiten Kennzeichnung versehen gespeichert. Mittels dem in dem Programmwort enthaltenen ersten Kennzeichen wird ein dem zugehörigen Primärinstruktionsteilwort entsprechendes Sekundärinstruktionsteilwort aus der zweiten Gruppe über das zugehörige zweite Kennzeichen ermittelt und das dem Programmwort entsprechende Sekundärinstruktionsteilwort erzeugt, indem in dem Sekundärinstruktionsteilwort aus der zweiten Gruppe die in dem Programmwort enthaltenen Instruktionsteile ausgetauscht werden.

Bereits bei der Konfigurationsphase können die Programmworte so aufgebaut werden, daß sie nur die geringstmögliche Information zur späteren Erzeugung eines Sekundärinstruktionsteilwortes in der Ausführungsphase beinhalten. Dies wird damit erreicht, daß auf die vorhergehenden Primärinstruktionsteile zurückgegriffen wird, die die größte Übereinstimmung mit dem zu konfigurierenden Primärinstruktionsteilwort aufweisen. Da in der Ausführungsphase die Sekundärinstruktionsteile in der gleichen Breite und in der gleichen Reihenfolge wie die Primärinstruktionsteile erzeugt werden sollen, wird in der Ausführungsphase praktisch in umgekehrter Reihenfolge zur der Konfigurationsphase verfahren und jeweils ein bereits erzeugtes Sekundärinstruktionsteilwort, welches dem Primärinstruktionsteilwort entspricht, das in der Konfigurationsphase einem aktuellen Primärinstruktionsteilwort am ähnlichsten ist, zur Erzeugung eines neuen Sekundärinstruktionsteilwortes verwendet wird. Da stets auf das Instruktionsteilwort mit der größten Ähnlichkeit oder der größten Übereinstimmung zurückgegriffen wird, läßt sich die Menge der zur Erzeugung eines neuen Sekundärinstruktionsteilwortes erforderliche Information so klein wie möglich halten. Damit kann der Speicherbedarf eines Programmspeichers minimiert werden.

In einer Ausführungsform des Verfahrens ist vorgesehen, daß die erste Gruppe aus einer ersten Anzahl der dem jeweiligen Primärinstruktionsteilwort direkt vorausgehenden Primärinstruktionsteilen besteht. Die zweite Gruppe besteht aus einer zweiten Anzahl von Sekundärinstruktionsteilen, die mindestens gleich der ersten Anzahl ist, wobei jedes Sekundärinstruktionsteilwort vor der Erzeugung des nächstfolgenden Sekundärinstruktionsteilwortes der Gruppe als letztes hinzugefügt und das über die zweite Anzahl hinausgehende zuerst hinzugefügte Sekundärinstruktionsteilwort aus der zweiten Gruppe entfernt wird. Damit besteht die Gruppe stets aus den unmittelbar vorhergehenden Instruktionsteilen, wovon eines eine größtmögliche Ähnlichkeit aufweist, wenn sich die Instruktionsteile innerhalb einer Gruppe inhaltlich unterscheiden.

Eine andere Möglichkeit der Ausgestaltung des Verfah-

rens besteht darin, daß das neu erzeugte Sekundärinstruktionswort der zweiten Gruppe hinzugefügt wird, indem dieses anstelle des seiner Erzeugung dienenden Sekundärinstruktionswortes gespeichert wird.

Mit dieser Variante des Verfahrens wird es ausgeschlossen, daß sich die Gruppe mit inhaltsgleichen Instruktionswörtern beim mehrmaligen Durchlaufen ein und derselben Funktion anfüllt.

Eine andere Möglichkeit besteht darin, daß das neu erzeugte Sekundärinstruktionswort nicht gespeichert wird. Dabei wird bei jeder Erzeugung eines Sekundärinstruktionswortes stets auf den gleichen Vorrat von einmal erzeugten Instruktionswörtern zurückgegriffen, was günstig ist, wenn diese gespeicherten Instruktionswörter als Musterwörter für die Erzeugung anderer Sekundärinstruktionswörter geeignet sind, dabei hierbei Speichervorgänge vermieden werden können.

In einer Ausgestaltung des Verfahrens ist vorgesehen, daß das erste Kennzeichen als Abstand des zum jeweiligen Programmwort gehörigen Primärinstruktionswortes zu dem Primärinstruktionswort mit der größten Übereinstimmung gebildet wird.

Eine weitere Ausgestaltung des Verfahrens sieht vor, daß das zweite Kennzeichen aus einer dem ersten Kennzeichen entsprechenden Adresse eines vorhergehenden Sekundärinstruktionswortes in einem der Abspeicherung der zweiten Gruppe dienenden Speicher besteht.

Diese beiden Ausgestaltungen zeigen eine besonders einfache Erzeugung von Merkmalen.

Eine Reduzierung der Programmwortbreite und damit verbunden eine Verringerung der Breite des Programmspeichers kann dadurch erreicht werden, daß das Programmwort aus einer Anzahl von Instruktionswortteilen besteht, die der innerhalb der Konfiguration am häufigsten auftretenden Anzahl zu unterscheidender Instruktionswortteile entspricht und daß zur Zusammensetzung von Sekundärinstruktionsworten, die zu dem der Erzeugung dienenden Sekundärinstruktionswort mehr als die in einem Programmwort abgelegte Anzahl von Instruktionsworten benötigt, mehrere Programmwörter verwendet werden. Somit genügt die Breite des Programmspeichers den häufigsten Anwendungsfällen, wobei auch keine Verzögerung in der Erstellung des Sekundärinstruktionswortes eintritt. Lediglich in den relativ wenigen Fällen, in denen die Anzahl der zu ändernden Instruktionswortteile die Breite eines Programmwortes überschreitet, werden zwei oder mehr Programmwörter in zwei oder mehr Takten zur Herstellung des Sekundärinstruktionswortes benötigt.

Eine weitere Verringerung der Programmwortbreite kann dadurch erreicht werden, daß die Instruktionswortteile in einem Programmwort komprimiert werden. Dies geschieht dadurch, daß die Bitbreite in dem Maße verringert wird, wie eine Darstellung der am häufigsten vorkommenden Instruktionswortteile möglich ist, und daß bei einem Auftreten von Instruktionswortteilen, die zur Darstellung eine größere Bitbreite benötigen, mehrere Programmwörter verwendet werden.

Für eine derartige Komprimierung des Programmwortes ist es insbesondere zweckmäßig, daß die Breite der Instruktionswortteile in dem Programmwort halbiert wird und zur Darstellung der Instruktionswortteile ein oder zwei Programmwörter vorgesehen sind. Hat beispielsweise normalerweise das Programmwort eine Breite von 8 Bit, so stehen 256 Kombinationen zur Verfügung. Ein großer Teil dieser Kombinationen wird jedoch nur äußerst selten oder gar nicht benötigt. Dementsprechend kann das Programmwort mit einer Breite von 4 Bit vorgesehen werden, was für die 16 häufigsten Kombinationen zur Darstellung ausreichend ist.

Soll eine andere als eine dieser Kombination dargestellt werden, werden hierfür zwei oder mehr Programmwörter verwendet. Da dies jedoch nur sehr selten auftritt, ist die erreichte Einsparung an Speicherplatz für den Programmspeicher größer als der mögliche Zeitaufwand zur Erzeugung seltener Kombinationen.

Die Aufgabenstellung wird erfindungsgemäß auch durch eine Prozessoranordnung gelöst, die dadurch gekennzeichnet ist, daß der Instruktionswortpuffer aus einem Speicher mit wahlfreiem zeilenweisen Zugriff besteht. Im Unterschied zu dem dargestellten Stand der Technik besteht damit die Möglichkeit, direkt auf verschiedene abgespeicherte Instruktionswörter zur Herstellung eines neuen Sekundärinstruktionswortes zurückzugreifen. Damit muß nicht irgendein gespeichertes Instruktionswort der Erzeugung eines neuen Sekundärinstruktionswortes dienen, sondern es kann dasjenige abgespeicherte Instruktionswort verwendet werden, das mit dem zu erzeugenden Sekundärinstruktionswort die größte Übereinstimmung aufweist. Damit ist der Erzeugungsaufwand relativ klein, was nur eine geringe Bitbreite des Programmwortes und damit einen kleinen Programmspeicher erfordert.

Die Erfindung soll nachfolgend anhand eines Ausführungsbeispiels näher erläutert werden. In den zugehörigen Zeichnungen zeigt

Fig. 1 ein Ablaufschema des erfindungsgemäßen Verfahrens und

Fig. 2 das Prinzip der erfindungsgemäßen Erzeugung der Primär- und Sekundärinstruktionswörter

Bei dem in Fig. 1 dargestellten Verfahren zur Ansteuerung von Funktionseinheiten 1 in einem Prozessor 2 wird aus einem Programmcode 3 mittels einer Übersetzung 4 in einer Konfigurationsphase eine Folge von Primärinstruktionsworten 5 aus mehreren Instruktionswortteilen 6 erzeugt. Diese Folge von Primärinstruktionsworten 5 wird komprimiert und als Folge von zugehörigen Programmworten 7 in einem Programmspeicher 8 gespeichert.

Die Programmwörter 7 bestehen aus einer Anzahl von Instruktionswortteilen 6, die der innerhalb der Konfiguration am häufigsten auftretenden Anzahl zu unterscheidender Instruktionswortteile 6 entspricht. Zur Zusammensetzung von Sekundärinstruktionsworten 9 können mehr als die in einem Programmwort 7 abgelegte Anzahl von Instruktionswortteilen 6 benötigt werden. Dazu werden dann mehrere Programmwörter 7 verwendet.

In einer nachfolgenden Ausführungsphase werden aus der Folge von Programmworten 7 aufeinanderfolgende Sekundärinstruktionswörter 9 aus mehreren Instruktionswortteilen 6 in der zur Ansteuerung aller Funktionseinheiten 1 benötigten vollen Instruktionswortbreite erzeugt.

Im Ergebnis der Konfiguration weist ein Programmwort 7 ein erstes Kennzeichen 10 eines Primärinstruktionswortes 5 aus einer ersten Gruppe 11 vorhergehender Primärinstruktionswörter 5, welches mit dem zum Programmwort 7 zugehörigen Primärinstruktionswort 5 die größte Übereinstimmung aufweist, und Instruktionswortteile 6, die das zu dem Programmwort 7 gehörige Primärinstruktionswort 5 von dem zu dem Kennzeichen gehörigen Primärinstruktionswort 5 unterscheidenden, enthält. Das erste Kennzeichen 10 wird als Abstand des zum jeweiligen Programmwort 7 gehörigen Primärinstruktionswortes 5 zu dem Primärinstruktionswort 5 mit der größten Übereinstimmung gebildet.

In der Ausführungsphase wird eine der ersten Gruppe 11 in der Anzahl entsprechende zweite Gruppe 12 von Sekundärinstruktionsworten 9 mit je einem zweiten Kennzeichen 13 versehen gespeichert. Das zweite Kennzeichen 13 wird aus einer Zeilennummer eines der Abspeicherung der zweiten Gruppe 12 dienenden Speichers 14 gebildet.

Mittels dem in dem Programmwort 7 enthaltenen ersten Kennzeichen 10 wird ein dem zugehörigen Primärinstruktionswort 5 entsprechendes Sekundärinstruktionswort 9 aus der zweiten Gruppe 12 über das zugehörige zweite Kennzeichen 13 ermittelt. Das dem Programmwort entsprechende Sekundärinstruktionswort 9 wird erzeugt, indem in dem Sekundärinstruktionswort 9 aus der zweiten Gruppe 12 die in dem Programmwort 7 enthaltenen Instruktionswortteile 6 ausgetauscht werden.

Die erste Gruppe 11 besteht aus einer ersten Anzahl der dem jeweiligen Primärinstruktionswort 5 vorausgehenden Primärinstruktionswörtern 5. Die zweite Gruppe 12 besteht aus einer zweiten Anzahl von Sekundärinstruktionswörtern 9, die mindestens gleich der ersten Anzahl ist, wobei jedes Sekundärinstruktionswort 9 vor der Erzeugung des nächstfolgenden Sekundärinstruktionswortes 9 der zweiten Gruppe 12 als letztes hinzugefügt werden kann. Das über die zweite Anzahl hinausgehende zuerst hinzugefügte Sekundärinstruktionswort 9 wird aus der zweiten Gruppe 12 entfernt.

#### Bezugszeichenliste

- 1 Funktionseinheit
- 2 Prozessor
- 3 Programmcode
- 4 Übersetzung
- 5 Primärinstruktionswort
- 6 Instruktionswortteil
- 7 Programmwort
- 8 Programmspeicher
- 9 Sekundärinstruktionswort
- 10 erstes Kennzeichen
- 11 erste Gruppe
- 12 zweite Gruppe
- 13 zweites Kennzeichen
- 14 Speicher

#### Patentansprüche

1. Verfahren zur Ansteuerung von Funktionseinheiten in einem Prozessor, bei dem in einer Konfiguration eine aus einer Übersetzung eines Programmcodes stammende Folge von Primärinstruktionsworten aus mehreren Instruktionswortteilen komprimiert und als Folge von zugehörigen Programmworten gespeichert wird und bei dem in einer nachfolgenden Ausführungsphase aus der Folge von Programmworten aufeinanderfolgende Sekundärinstruktionsworte aus mehreren Instruktionswortteilen in der zur Ansteuerung aller Funktionseinheiten benötigten vollen Instruktionswortbreite erzeugt werden, **dadurch gekennzeichnet**, daß im Ergebnis der Konfiguration ein Programmwort (7) ein erstes Kennzeichen (10) eines Primärinstruktionswortes (5) aus einer ersten Gruppe (11) vorhergehender Primärinstruktionsworte (5), welches mit dem zum Programmwort (7) zugehörigen Primärinstruktionswort (5) die größte Übereinstimmung aufweist, und Instruktionswortteile (6), die das zu dem Programmwort (7) gehörige Primärinstruktionswort (5) von dem zu dem ersten Kennzeichen (10) gehörigen Primärinstruktionswort (5) unterscheidenden, enthält, daß in der Ausführungsphase eine der ersten Gruppe (11) in der Anzahl entsprechende zweite Gruppe (12) von Sekundärinstruktionsworten (9) mit je einem zweiten Kennzeichen (13) versehen gespeichert wird, und daß mittels dem in dem Programmwort (7) enthaltenen ersten Kennzeichen (10) ein dem zugehörigen Primär-

instruktionswort (5) entsprechendes Sekundärinstruktionswort (9) aus der zweiten Gruppe (12) über das zugehörige zweite Kennzeichen (13) ermittelt wird und das dem Programmwort (7) entsprechende Sekundärinstruktionswort (9) erzeugt wird, indem in dem Sekundärinstruktionswort (9) aus der zweiten Gruppe (12) die in dem Programmwort (7) enthaltenen Instruktionswortteile (6) ausgetauscht werden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die erste Gruppe (11) aus einer ersten Anzahl der dem jeweiligen Primärinstruktionswort (5) direkt vorausgehenden Primärinstruktionswörtern (5) besteht und daß die zweite Gruppe (12) aus einer zweiten Anzahl von Sekundärinstruktionswörtern (9) besteht, die mindestens gleich der ersten Anzahl ist, wobei jedes Sekundärinstruktionswort (9) vor der Erzeugung des nächstfolgenden Sekundärinstruktionswortes (9) der zweiten Gruppe (12) als letztes hinzugefügt und das über die zweite Anzahl hinausgehende zuerst hinzugefügte Sekundärinstruktionswort (9) aus der zweiten Gruppe (12) entfernt wird.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das neu erzeugte Sekundärinstruktionswort (9) der zweiten Gruppe (12) hinzugefügt wird, indem dieses anstelle des seiner Erzeugung dienenden Sekundärinstruktionswortes (9) gespeichert wird.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das neu erzeugte Sekundärinstruktionswort (9) nicht gespeichert wird.

5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß das erste Kennzeichen (10) als Abstand des zum jeweiligen Programmwort gehörigen Primärinstruktionswortes zu dem Primärinstruktionswort mit der größten Übereinstimmung gebildet wird.

6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß das zweite Kennzeichen (13) aus einer dem ersten Kennzeichen entsprechenden Adresse eines vorhergehenden Sekundärinstruktionswortes in einem der Abspeicherung der zweiten Gruppe (12) dienenden Speichers (14) besteht.

7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß das Programmwort (7) aus einer Anzahl von Instruktionswortteilen (6) besteht, die der innerhalb der Konfiguration am häufigsten auftretenden Anzahl zu unterscheidender Instruktionswortteile (6) entspricht und daß zur Zusammensetzung von Sekundärinstruktionsworten (9), die zu dem der Erzeugung dienenden Sekundärinstruktionswort (9) mehr als die in einem Programmwort (7) abgelegte Anzahl von Instruktionsworten (9) benötigt, mehrere Programmwörter (7) verwendet werden.

8. Verfahren nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, daß die Instruktionswortteile (6) in einem Programmwort (7) komprimiert werden, indem die Bitbreite in dem Maße verringert wird, wie eine Darstellung der am häufigsten vorkommenden Instruktionswortteile (6) möglich ist, und daß bei einem Auftreten von Instruktionswortteilen (6), die zur Darstellung eine größere Bitbreite benötigen, mehrere Programmwörter (7) verwendet werden.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, daß die Breite der Instruktionswortteile (6) in dem Programmwort (7) halbiert wird und zur Darstellung der Instruktionswortteile (6) ein oder zwei Programmwörter (7) vorgesehen sind.

10. Prozessoranordnung zur Durchführung des Verfahrens nach dem PA1 mit Funktionseinheiten, einem

diesen Funktionseinheiten zugeordneten Instruktions-  
wortspeicher und einem Instruktionswortpuffer zur  
Speicherung bereits erzeugter Instruktionsworte mit ei-  
ner Breite, die mindestens gleich der Bitbreite des Se-  
kundärinstruktionswortes ist, dadurch gekennzeichnet, 5  
daß der Instruktionswortpuffer aus einem Speicher (14)  
mit wahlweisem zeilenweisen Zugriff besteht.

---

Hierzu 2 Seite(n) Zeichnungen

---

10

15

20

25

30

35

40

45

50

55

60

65

**BEST AVAILABLE COPY**

- Leerseite -

**BEST AVAILABLE COPY**

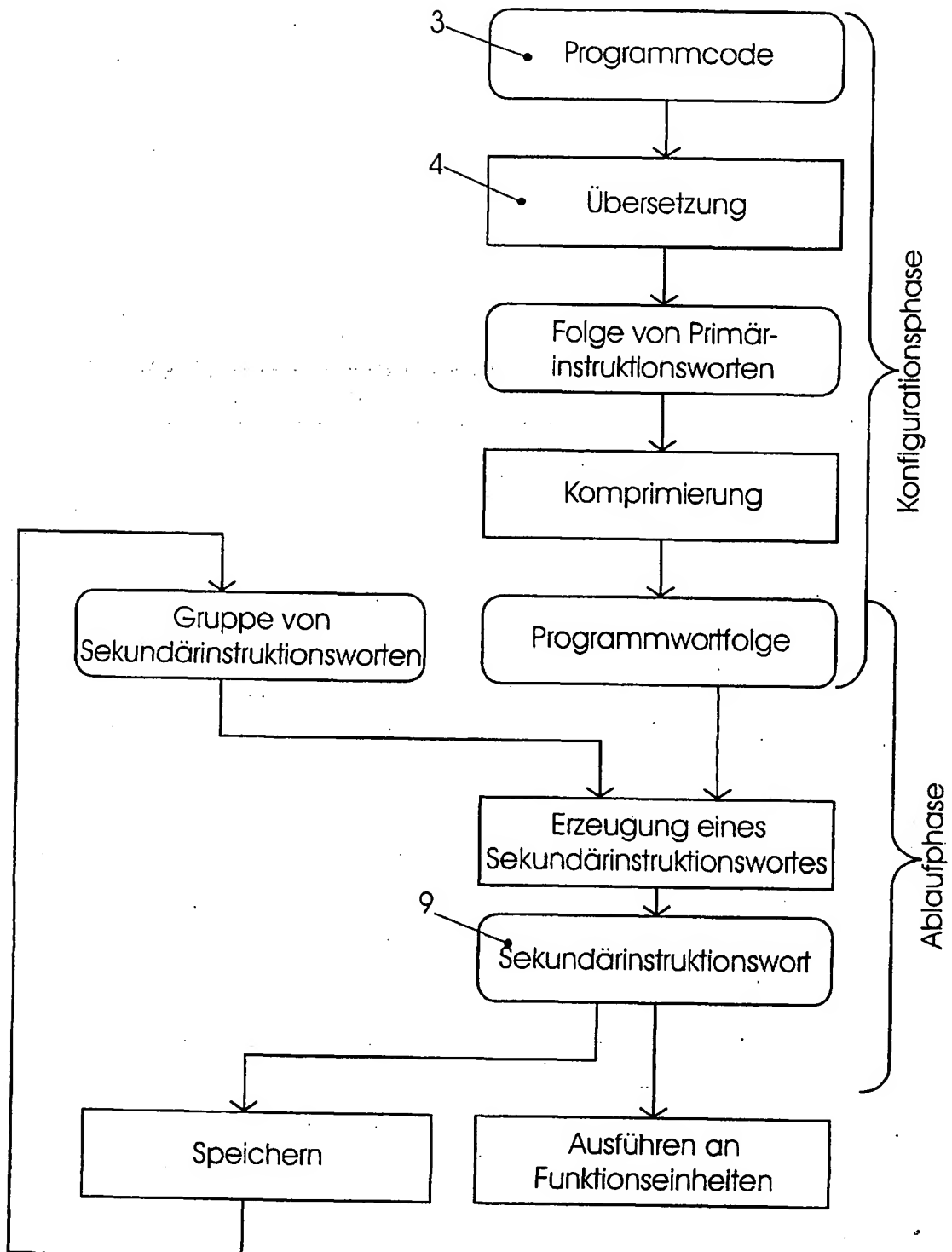
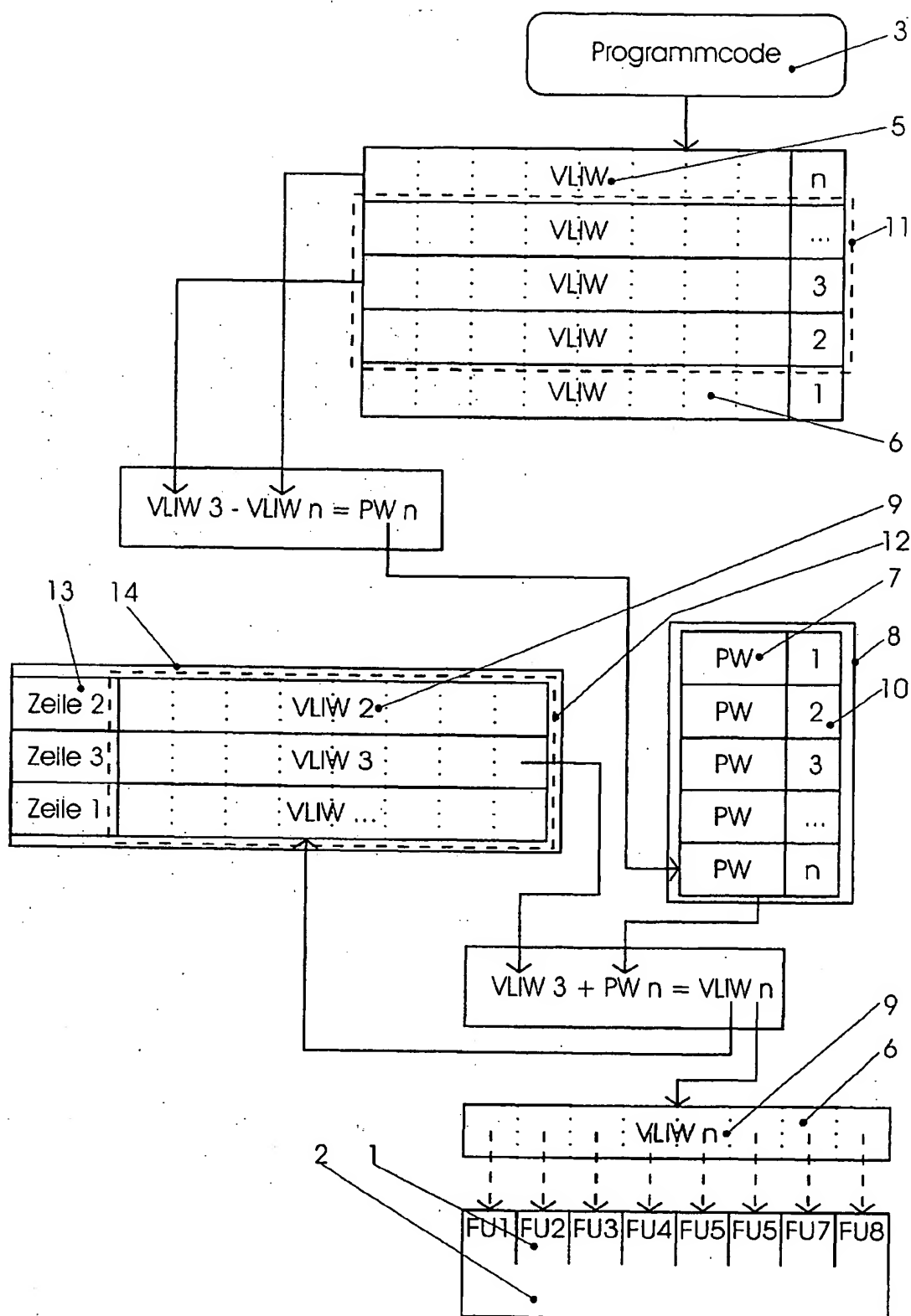


Fig. 1

BEST AVAILABLE COPY



BEST AVAILABLE COPY

Fig. 2